Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
01 OCT 2014	**N/A**	**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Probabilistic Analysis of Time Sensitive Systems	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**SAR**	**1**	

Probabilistic Analysis of Time Sensitive Systems

Problem Statement

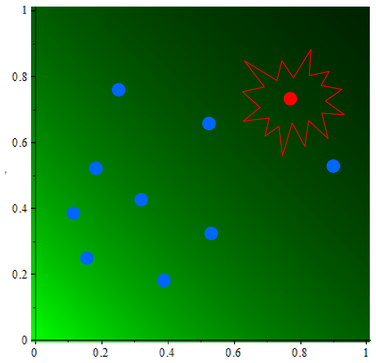
Time-sensitive systems in uncertain environments have complex behaviors. How do we assure correctness of such systems?

- Exact probabilistic verification is infeasible due to model size
- Black box testing does not yield bounded predictions
- Need formal approach for dealing with uncertainty
- Accurate, bounded, probabilistic results
- In reasonable time even for rarely occurring errors

Stochastic Model Checking (SMC)

SMC is a rigorous simulation-based approach for estimating that a property holds in a system.

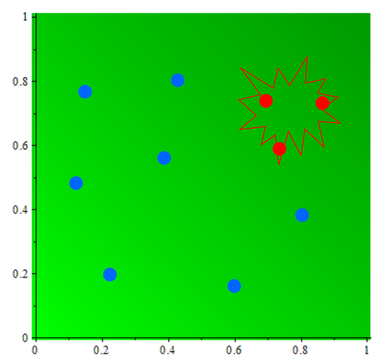
- System properties described in formal language (BLTL, etc.)
- Property is tested on "sample trajectories" (sequence of states)
- Each outcome treated as a Bernoulli trial (i.e., coin flip)



SMC Basics

- Indicator function $I(\vec{x}) = 1$ iff property holds for input $\vec{x}$.
- Relative Error $RE(\hat{p}) = \frac{\sqrt{var(\hat{p})}}{E[\hat{p}]}$ is measure of accuracy.
- Draw random samples from input distribution $f(\vec{x})$ until target Relative Error is met.
- Estimated probability that property holds is:

$$\hat{p} = \frac{1}{N}\sum_{i=1}^{N} I(\vec{x}_i) = \frac{1}{10} = 0.1 \qquad \text{RE}(\hat{p}) = \frac{0.32}{0.1} = 3.2$$



Importance Sampling

- Modify input distribution to make rare properties more visible.
- Weighting function $W(\vec{x})$ maps solution back to original problem.
- Reduced relative error with same number of samples.

$$\hat{p} = \frac{1}{N}\sum_{i=1}^{N} I(\vec{x}_i)W(\vec{x}_i) = \frac{0.2 + 0.5 + 0.3}{10} = 0.1$$

$$\widehat{RE} = \frac{0.18}{0.1} = 1.8$$

Semantic Importance Sampling
A New Approach to Importance Sampling

Input Specification in C

```
#include "osmosis_client.h"

//@dist a=uniform(min=0,max=5)
//@dist b=normal(mean=3,std=1,min=0,max=5)
void simple()
{
  double a = INPUT_D("a");
  double b = INPUT_D("b");
  double c = a + b;
  double d = (a – b)/2.0;

  ASSERT(sin(c)*cos(d/2) < 0.995);
}
```

Translate C model to SMT2 for Analysis.

SMT2 Model

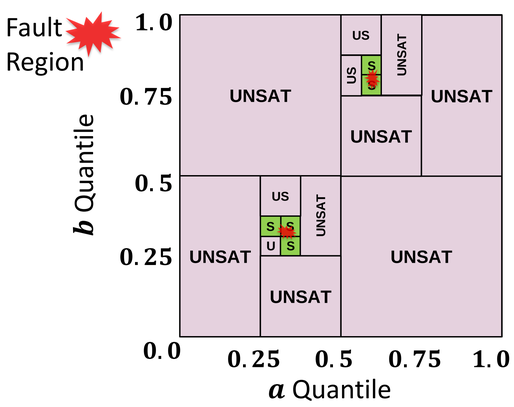
```
(set-logic QF_NRA)
(declare-fun a () Real)
(declare-fun b () Real)
(declare-fun a_1 () Real)
(declare-fun b_1 () Real)
(declare-fun c_1 () Real)
(declare-fun d_1 () Real)
(assert (>= a 0))
(assert (<= a 5))
(assert (>= b 0))
(assert (<= b 5))
(assert (= a_1 a))
(assert (= b_1 b))
(assert (= c_1 (+ a_1 b_1)))
(assert (= d_1 (/ (- a_1 b_1) 2.0)))
(assert (not (< (* (sin c_1) (cos d_1)) 0.9)))
(check-sat)
(exit)
```

Input Cube

ASSERT()

Recursively invoke dReal SMT checker to build abstract model of specification.

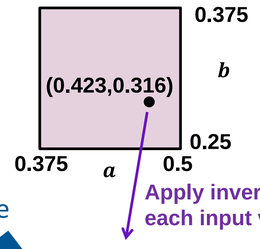
Abstract Indicator Function $I^*(\vec{x})$



Use $I^*(\vec{x})$ to generate random input vectors:

- Randomly pick SAT cube
- Randomly pick point in cube

Input Generation



Apply inverse CDF on each input variable.

$(a, b) = (2.115, 2.503)$

Apply generated inputs to original C model to calculate bounded failure probability estimate.

Raw Prob. Estimate

$$\hat{p}_{raw} = 0.024$$

$$RE(\hat{p}_{raw}) = 0.01$$

Weight function $W(\vec{x}_i)$ is probability p^* that $\vec{x}$ is in $I^*(\vec{x})$.

Abstract Probability

$$p^* = \frac{5}{2^8}$$

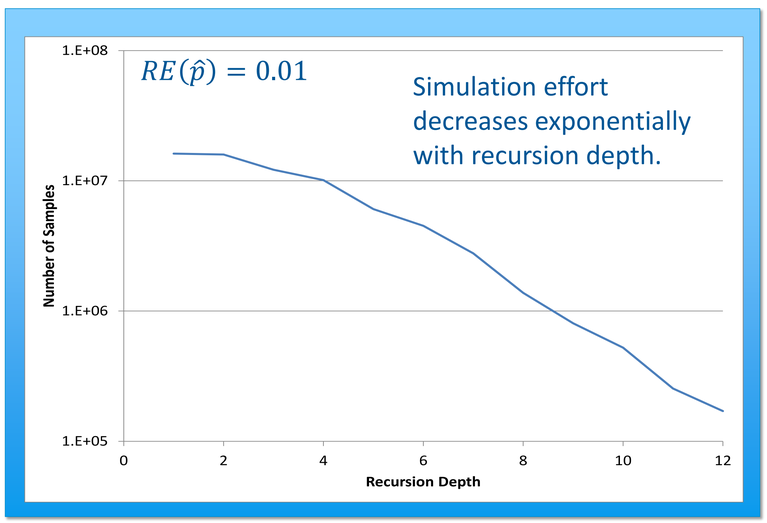
Number of cubes in $I^*(\vec{x})$.

Level of cubes

Final Probability Estimate

$$\hat{p} = p^*\hat{p}_{raw} = 0.00047$$

$$RE(\hat{p}) = 0.01$$



Software Engineering Institute | Carnegie Mellon University

Jeffery Hansen – jhansen@sei.cmu.edu

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM-0001712